SEPG
FEDERAL AVIATION ADMIN
INTERFACE
Newsletter of the Software Engineering Process Group
Volume 5, Number 3 • August 1996

Engineering Disasters...

by Norman F. Simenson, AIT-5

Engineering disasters are ***ALWAYS*** due to bad management and ***NEVER*** to bad engineering because, for any but the smallest noncritical project, the Program manager must do risk management. The plea "failed due to uncontrollable forces," or, "my team let me down," is almost invariably an admission of bad management. Worse, it means not only that the manager was inept, but that she doesn't even have a clue as to how to improve. A manager is not simply an overpaid administrator who handles the budget and shares out the work. A Program manager is the principal defense against bad engineering and other real-life disasters which can destroy any Program!

Simply put, the manager must expect and plan for bad engineering or other engineering problems. This includes workable contingency plans which will prevent bad engineers or unexpected technical difficulties from impacting schedule and cost. Even with the best of engineers, creativity cannot be planned for and scheduled like a train time table. The most mundane engineering projects contain requirements for a substantial degree of good engineering and/or creativity.

There are all sorts of contingency plans and safeguards against problems due to bad engineering. Some will also work for almost any unexpected technical difficulties. The two categories are different, and should be treated differently, but have some overlap and similarities for planning purposes. Here, we will only take a superficial view of planning to prevent the impact of bad engineering.

The first line of defense for the manager is correctly assessing the abilities of and assigning the engineers to do a job. The "hall" technique of assigning people is a sure recipe for disaster. This technique assumes that everyone within the same labor category is equivalent. So, it must be okay to assign the first warm body of the appropriate category to pass the manager's door in the hallway to the job at hand. The amazing thing is not how often the warm body fails but, rather, how often it succeeds!

For any given job, there are four types of people. One type has never done a like job, and may possess some or most of the necessary skills but is fundamentally an unknown quantity. The other three types have done a like job before,

continued on page 7

Surprise!!!

by Robert N. Charette,

Teddy Roosevelt once said that risk is like fire—controlled, it will help you; uncontrolled it will rise up and destroy you. As individual information systems become ever larger, more complex, absurdly expensive, and burdened with increasing consequences of operational failure, it is becoming abundantly clear to even the most hidebound skeptic that taking actions proactively to controlling the risks involved in acquiring, developing, and operating information systems are no longer optional.

A critical component in the process of controlling risk is the act of communication—the public admission of a risk's existence so that action can be taken before the risk turns into a problem, or worse, a crisis. One of the complaints I hear repeatedly from project managers when performing risk assessments is that they are endlessly ambushed by problems. Problems that, upon inspection, they could have easily intervened to eliminate altogether—***if***, that is, they had been told about them while they were still ***potential*** problems. What with getting surprised being the one thing project managers hate more than anything, it is a sad commentary that the lament, "If I had only known!" continues to be a too familiar refrain.

By the time they suffer their second or third "surprise," many project managers begin to harbor a deep suspicion that their staff is either chronically lying to them, are incompetent clowns, or both. "Why," they ask," won't anyone tell me completely and honestly what is going on here?" I frequently try to explain to them that the main reason is because their project staff is most likely afraid of speaking openly about the risks they may perceive. It often seems safer to allow risks to fester, hoping they will go away or can be controlled with limited resources, letting them grow into serious problems rather than taking positive action to mitigate them and/or making them public.

"Preposterous!," is the usual reply, "my door is open to everyone." Unfortunately, my risk assessments usually find

continued on page 5

WHEN THE WAGES OF POOR PLANNING ARE DEATH

"The causes of the disaster are not due to faulty organisation, but to misfortune in all risks which had to be undertaken.... We took risks, we knew we took them; things have come out against us, and therefore we have no cause for complaint, but bow to the will of Providence, determined still to do our best to the last..." These words are from the last message of the British explorer Robert Falcon Scott, dated March 29, 1912, which concluded, "Every day we have been ready to start for our [One Ton] depot 11 miles away but outside the door of the tent it remains a scene of whirling drift. We shall stick it out to the end, but are getting weaker, of course, and the end can not be far. It seems a pity but I do not think I can write anymore."

The storm was to last nine days, frustrating a second relief attempt from their base on the 30th. Not until November 12, 1912 were Scott's diaries, letters, and last message discovered with his frozen body and those of his two remaining companions. Another companion had committed suicide on March 17, 1912 so as not to be a burden and still another had died of injuries and complications of malnutrition a month after the start of the return trip.

Starting in the last week of October, 1911, Robert Scott and Roald Amundsen had raced for the South Pole. Amundsen arrived at the Pole on December 14, 1911; Scott did not arrive until January 16, 1912. On the return, Scott and his remaining party ran out of food and died just short of their last depot. The image of the doomed Scott giving his all in a vain effort to be first to the Pole was to overshadow Amundsen's achievement for the remainder of the Norwegian's life. Amundsen, notorious for his attention to every detail, was derisively dubbed the "professional" by the British. Scott was the glorious amateur, never stopping to reckon the odds.

Since Amundsen's feat had been achieved largely due to the use of 52 North Greenland Huskies (which also served as food for the remaining huskies when no longer needed), the conventional wisdom was that Scott died only because he could not bear the seeming cruelty of using dogs to pull cargo sleds and because he ran into unexpectedly bad weather (although he had planned to be out for some two months later than Amundsen).

But in 1979, Scott's image was brutally debunked by Roland Huntford in his book, ***The Last Place on Earth***. Scott is mercilessly revealed to have died because of repeated errors of judgment and a failure of leadership. As Huntford put it, "The Scott of [his] diaries is rash, then rueful; timid and dangerously reckless by turns; palpably lacking in judgment; uncertain, indecisive, confused by emergencies, incapable of learning from experience; totally lacking in foresight, and trusting to luck. [His] judgment is consistently clouded by emotion [and prejudice]; and ... inhabit[s] the borderland between illusion and reality." Standards change. In 1912, the valiant effort was everything—as one commentator put it, "better a dead lion than a live donkey." In 1979, the useless waste of five lives was unforgivable.

Through sometimes near fatal experiences, most of us learn to build a wide margin of safety into planning for all high risk activities, such as mountain climbing or open ocean sailing. A mountain top or mid-Atlantic is not provided with convenience stores where one can easily replace any forgotten, misplaced, or defective article which may be necessary for survival. Perhaps the risk that a bad storm at sea will leave you at the ocean bottom is much less than one thousandth of one-percent, but improbable events have a way of catching up with you if you tempt fate long enough—such as by not checking your fuel supply and the weather report each time you put to sea.

Scott's diaries reveal a regular failure to provide any margin of safety. He found the detailed planning and the work necessary for an expedition boring and had difficulty in assigning work, or in making intelligent use of the work or experience of others. Suggestions were invited, but were ignored or badly received. Eventually they ceased, except for protests. He was careless in a place where carelessness equated to recklessness.

Scott ran out of supplies 11 miles short of his One Ton Depot, just one day's travel in clear weather, at the end of a 1500 mile, five month journey. The depot's location had resulted from dozens of poor decisions. But he still may have made it if he hadn't dawdled during the early stages of the return, when numerous indications should have alerted him that he was already seriously short of supplies and that his men had begun to suffer the early stages of scurvy and other deficiency diseases. In similar circumstances in a 1908 expedition, Ernest Shackleton quit just 87 miles short of the Pole rather than needlessly risk lives.

Over thirteen months earlier, Scott had been strongly urged to put the One Ton Depot at least 21 miles further south. (Amundsen cached three tons up to 150 miles further south.) Scott refused. The supplies that he cached in his other depots were insufficient, even by the most optimistic of calculations. (Amundsen actually left supplies behind and still gained weight on the round trip to the Pole.) Scott was indifferent to the fact that the eventual leader of the first relief party from the base camp was unable to master navigation and so dared not venture beyond the One Ton Depot for fear of missing Scott and himself getting lost. Scott never bothered to mark his actual route of travel, which slowed him on his return and also discouraged the first relief party from venturing along his trail. His last orders for relief of the Polar party were unclear, ambiguous, unnecessarily restrictive, and contradictory. Scarcely more than a one week march and clear weather separated the Polar and relief parties on March 10.

Although Scott started with dogs, Manchurian ponies, and motorized sleds, in the

*inter*FACE

is published quarterly by SEPG

DOT/FAA/AIT-5
800 Independence Avenue, SW
Washington, DC 20591

Chief Scientist for Software Engineering
Floyd Hollister (202) 267-8020

Editor
Norm Simenson (202) 267-7431

FAX (202) 267-5080

end, he and his men manually hauled their sleds up 10,000 feet in elevation and over 1000 miles to the Polar plateau and back. Against all evidence, Scott felt the dogs incapable of the trek to the Pole, the ponies died or were shot, being ill-suited to the climate and conditions, and the motorized sleds were lost, largely because of lack of tools and spare parts, imperfectly trained mechanics (an experienced Polar explorer and co-inventor of the sleds had been dropped from the expedition on little more than whim), and carelessness.

Scott's Polar party were on skis and foot, but the men on skis had only learned to use them during the current expedition. (Amundsen's men were all on skis, had had years of experience on skis, and dogs were used to pull the sleds and often the men on skis. They averaged more than 50% more miles, including one day of rest in five, in one third to half as many hours per day of travel.) Scott's men additionally burdened themselves with hundreds of pounds of useless scientific instruments, and rocks—"interesting geological specimens"—gathered along the way. Thirty pounds of rocks from the Beardmore glacier were found among Scott's remains. Fifteen pounds of pemmican would have saved them at the end. There were dozens of other instances of poor planning and decisions. Ordinary prudence dictated allowing for a very wide margin of safety. He left none.

Scott failed at more than allowing a sufficiently wide margin for emergencies. He consistently ignored the advice of others, often far more skilled at their specialties. He drove his men to the point of exhaustion and collapse, often pointlessly. Several members of his team with critical skills either quit in disgust early in the expedition, or were incapacitated at crucial moments because of valiant attempts to carry out Scott's ill-conceived orders. He was notorious for deferring planning until the last possible moment and for making major last minute alterations to existing plans, sometimes repeatedly. Scott's ultimately fatal alteration of plans may have been to change the Polar party to five men at the very last minute when all preparations had been for four. He seemed to make the whole Polar expedition into a sporting contest between him and his men and between him and the Pole. The Pole won.

For sixteen months before his death, Scott's diaries repeat endlessly how perfect his preparations were, but document how deficient these were in fact. On his previous Antarctic expedition in 1901 to 1904, Scott barely survived the same errors that cost him his life in 1912: sparse and ill-marked depots, difficulties with his companions, a diet that led to scurvy, last minute improvisation and planning (and changes to plans), and no margin of safety. But he was adept at convincing himself and others that the fault always lay elsewhere. Anyone who habitually seeks fault elsewhere deprives himself of the chance to learn from mistakes and is condemned to repeat them.

This narrative has nominally been about Scott's and Amundsen's very different styles in the Antarctic. But the chief problems each handled in their very different way were those of calculating and making proper provision for risks, motivating others, and accepting ultimate responsibility. These are problems of everyday life. If we are careless about risks, we end up blighting our lives and those of others with a business failure, or an easily preventable accident or illness. Which program manager do you want—the bold adventurer and gambler, or the careful planner and skillful innovator? Risk management is all about not having to improvise at the last minute. ■

Letter from the EDITOR

A RISKY HISTORY

For all of you out there who think Risk Management is just the latest deviltry invented by the Harvard School of Business to harass the working troops, it no doubt will come as something of a shock that Risk Management was practiced by the Phoenicians some three or four thousand years ago in the form of ship's cargo or voyage insurance.

The insurance industry, whose stock in trade is risk management, is one of the oldest, consistently profitable industries in history. It has managed this feat despite being held liable for all kinds of unanticipated events and catastrophes. We can all learn from its history about how to reduce or avoid the impact of unhappy events. Indeed, I would imagine that almost all of us have health insurance, life insurance, etc. How about Program insurance?

Well, for the most part IPTs will have to self-insure, which means learning enough about risk management to apply it effectively.

I hope you like the lead article "Engineering Disasters..." originally entitled, "Why Engineering Disasters are ***ALWAYS*** due to Bad Management and ***NEVER*** to Bad Engineering." It was inspired by a manager who informed me that, "You engineers never take responsibility for the bad engineering which results in engineering disasters!" It is all about Ultimate responsibility and the fact that Program and project managers must first of all be risk managers. It is the job of the Program manager to insure the Program against all risks, including bad engineering. If there were no risks, we would not need Program managers. In fact, too many businesses are now trying just that, and rediscovering (the hard way) why Program managers are needed. Don't make that mistake as an IPT. The role of PM need not be performed by a designated individual, but the need for the PM's value added will not go away.

All in all, I consider this one of our more inspired issues! **Do** read and respond to all of the articles. We welcome letter's to the editor on topics in our newsletter or on any subject. Just send me an eMail...

Norm

FAA SOFTWARE ENGINEERING PROCESS GROUP

Floyd Hollister SEPG Chairperson	AIT-5
Susan Gardner SEPG Leader	AIT-5
Malcolm Andrews	AUA-500
Roger Cooley	AIT-5
Ray DeCerchio	ASU-120
Rebecca Deloney	AOS-5
Cheryl Dorsey	AIR-130
Matoka Forbes	AUA-400
Stewart Gibb	AND-460
Rob Hanes	AUA-200
Ed Harras	ASD-110
Susan Houston	AND-460
Linda Ibrahim	AIT-5
Paul D. Johnson	AUA-540
Cindy King	AUA-220
Ken Kraus	AND-430
Tom Marker	ASU-200
Larry Nivert	AND-620
Joel Petersen	AND-530
Dave Reusser	AIT-10
Ross Ridgeway	AMI-100
Dave Robinson	AIT-200
Debbie Rooney	AUA-540
Bruce Siebenthall	AND-520
Richard Simon	AND-130
Vivian Smith	AUA-600
Marie Stella	AND-8
Carolyn Strano	ASD-120
Richard Turner	ASD-200

FACTORS THAT INFLUENCE PROGRAMMATIC RISK

by Gary Preckshot, Ed Jones, and Dennis Lawrence
Lawrence Livermore National Laboratory

When making important personal or business decisions, we must all evaluate the risks of each alternative and weigh the trade-offs between one type of risk and another.

In many business and technical areas, a considerable body of analytical and experiential data and tools are available which can provide considerably better results than the heuristics or intuition derived from a single individual's experience. Program managers tend to be familiar with the concepts of risk assessment, analysis, and mitigation in selected areas such as human factors, security, and safety. But a formal risk-based approach provides traceability, control, and mitigation strategies and tactics applicable to most areas of decision-making.

When acquiring non-trivial software systems, program managers can use this same risk-based approach to procure a system that works correctly, is delivered on time and within budget, and meets the needs of the organization. Managers can assess specific problems or issues, rank their relative importance, determine which have the greatest payoff potential for management intervention, allocate resources appropriately, and monitor the effectiveness of the actions taken.

Each participant in a software acquisition is likely to calculate risk somewhat differently. For instance, the acquisition program manager who may be purchasing COTS software or software development services is not only concerned with meeting her organization's user needs, but also its goals and objectives, as determined by upper management. This may result in putting cost and schedule ahead of performance objectives. Meanwhile, the development program manager may view risk in terms of specific contractual conditions which may add to costs or schedule, late access to a market window, or an unprofitable level of maintenance effort. Regulators, on the other hand, focus almost exclusively on protecting workers and the public from unsafe products. Therefore, it is important to understand where one fits in this spectrum of risk perception, and to realize that one's viewpoint may not be shared by others involved in the software acquisition. The risk management activities of a software acquisition process must harmonize the differing objectives of the various participants, or at least keep them in rational balance. Otherwise, people will be working at cross-purposes and resources will be wasted or quality will suffer.

The acquisition program manager needs to understand the factors that influence software development risk as well as acquisition program risk. This is because, at least for developments special to the acquisition, the acquisition schedule and cost are closely linked to the development schedule and cost; anything which impacts the latter usually impacts the former. And, while a good acquisition program manager can take some steps to insulate the overall cost and schedule from those of the development, often there is an amplification effect as delay in one leg of a development chain impacts later stages. Anything which leads to an unstable development environment, including an unstable acquisition environment, will almost certainly be reflected in increased development schedule and costs. This usually leads to increased acquisition schedule and costs.

During 1992 and 1993, Lawrence Livermore National Laboratory (LLNL) carried out an investigation on behalf of the U.S. Nuclear Regulatory Commission designed to isolate criteria which can be used by ***regulators*** to assess the capability of organizations producing software for ***safety-critical*** applications. Data were obtained and analyzed from a variety of sources, including internationally recognized experts, leading companies, widely accepted software engineering standards, and the technical literature. One result was an organized list of "design factors" whose presence (or absence) provides useful clues to the capabilities of software developers. The factors were divided into four graded categories, the first and last of which are summarized below. (See "Design Factors for Safety-Critical Software," LLNL, for additional information.)

Seven factors are considered mandatory. The lack of any one of these factors may be considered sufficient grounds for rejection of a software product. These factors are:

- The availability of, effective use of, and sufficiency of high-quality management and technical personnel.
- The use of adequate and effective configuration management.
- The existence of clear, stable, and validated software requirements, with a well controlled change process.
- The use of a developer-independent organization for verification and validation of all program elements, including testing.
- The use of a formal, well-defined life cycle for product development and beyond.
- Traceability from user requirements, through system specification and design, through software requirements specification and design, through software code, and through unit, integration, and validation testing.
- The use of safety hazard analysis and risk analysis to guide development.

Nine factors were identified whose presence should be cause for caution and more thorough scrutiny. These negative factors are indicators of an organization in trouble.

- High staff turnover.
- A history of projects being driven by schedule rather than quality.
- Lack of a sufficiently long organization process history.
- Management which cannot (or does not) enforce stable requirements.
- A history of management's estimates of product reliability greatly exceeding actual experience.
- A history of failing to meet predicted cost, schedule, and quality goals for products.
- A failure to track errors and determine root causes.
- A current development effort that is underfunded (e.g., because of underbidding).
- A corporate culture that discourages problem-reporting by employees ("kill the messenger" syndrome).

Management of the procurement team is within the acquisition program manager's control, and may represent the single, most effective risk management tool available for reducing overall programmatic risk, provided there is adequate contingency planning. The "design factor" study done by LLNL, with slight modifications for local conditions, should prove to be a valuable management tool for most program managers.

An extensive compendium of LLNL's work on software reliability and software risk factors is available in Adobe Acrobat format at **http://nssc.llnl.gov/FESSP/CSRC/CSR.html**. LLNL references can also be found there. Please contact Gary Lynn Johnson, johnson27@llnl.gov, or the authors (preckshot1@llnl.gov, jones37@llnl.gov, or lawrence2@llnl.gov) for more information. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory. ■

*"**Surprise**" continued from page 1*

project managers creating numerous disincentives to open communication of project-related risk that they aren't even aware of. For example, how many times have they explicitly (or implicitly) told their project staff not to bring them problems, but only solutions instead?

Risks are inherently different from problems in that they pose real dilemmas that—in balancing resources and objectives—rarely have clear-cut "solutions." Thus, if the project manager really wants a solution, then the best strategy for a staffer may be to wait for the risk to turn into a problem. At that point, a clean solution can be much more easily found and the staffer does not have to worry about making hard choices between objectives and resources or perhaps guarding against a disaster which may not occur. If the staffer is lucky, the risk event may not even happen and no solution will be needed at all! The staffer has probably avoided some unnecessary unpleasantness and the project manager remains happy (in blissful ignorance)—everyone is happy!

The manager may want to consider whether staffers are subtly punished if risks which have been highlighted and guarded against, using (always) scarce resources, never come to pass. We all hate to pay insurance premiums for disasters which do not occur, but prudent people do not stop paying and do not punish their insurance agent for suggesting increases. We assume that that is the job of the insurance agent.

As a project manager, you need to be very careful about what you do or do not ask for, because you are most likely to get it. For instance, have you specifically asked for the project's risks in this week's meeting or status report? If not, why not? I bet you never forget to ask about the project's problems. Aren't risks also important enough to gain your attention?

Do you tend to "pooh-pooh" the risks identified, saying in effect that the person is making a mountain out of a molehill? Do you treat those who identify risks as if they were "nattering nabobs of negativism" who are not being team players? Do you implicitly equate the existence of risk with someone not doing a job properly by asking, "Who is responsible for this screw-up?"—implying someone must be to blame? Do you hide risks from the customer, in hopes that the risks may go away? Do you reward your "firemen" or "crisis managers" better than the project staff who quietly manage to do their jobs by avoiding "fires?" Do you spend most of your time visiting the successful parts of the project and find you have little time for those in trouble? Do you always cast a positive spin on any bad news, and expect everyone else to be always positive and upbeat?

An honest response to these questions will indicate how truly "open" a manager is to learning about risk, as well as how soon she is likely to get surprised again. To help reduce being constantly surprised by realized problems that quickly turn into full fledged crises, a manager needs to keep the following in mind.

Observe, first, that your project plans are not perfect. The estimating method used to create the plan in the first place operates under the "garbage-in, garbage out" principle, and the data used to create the plans are guaranteed to be uncertain to some degree. Absolute certainty is the prerogative of the Almighty only. (Don't make the mistake of placing yourself in That role.) The larger the project, the larger the degree of uncertainty. If you can't acknowledge that uncertainty exists in your plans, why expect your staff to? Second, readily acknowledge that, because of this uncertainty, **your** project, ***like all projects***, will be riddled with risks that won't ever be completely eliminated until the project is either completed or canceled. Plan ahead for bad engineering, creative solutions which don't arrive on schedule, etc.

Sell the idea of "active risk management" to your staff. Convince them that it is less costly to the project (and less wearing on you and them) to insure against risks up front than to confront major problems later. Openly encourage and support searching for and taking actions to mitigate project risks, early and often. Establish a regular, standard process for this risk search and mitigation activity—don't make it an exceptional event. Ask for risk status in your weekly project meetings. Encourage discussion and debate as to what alternative courses of action can be taken. Don't stop rewarding your "firemen"—you'll always need them—but also **reward those who successfully mitigate their risks early**. Recognize that decisions of whether to accept or mitigate risk, as well as problems, involve degrees of correctness, not right or wrong. If a decision turns out to be incorrect, don't lay blame. ***Do*** try to find out if the information used was too uncertain, whether the decision process itself was at fault, whether a wrong tactic or strategy was used, or whether it was in reality the best decision that could have been made at the time. Such post-mortems are important tools for improvement and ensure that a better decision may be made next time.

Get the customer involved, from the beginning if possible, since they are the key to controlling or eliminating many of your risks. Get bad news out early and don't hide risks from your customer. If you do, expect your staff to hide things from you. Amazingly, honesty ***is*** the best policy. It may be hard, especially at first, but it's all in your own self-interest.

It has been said, but bears repeating—project managers are first of all risk managers. If civilization can be said to have begun with man's control of fire—then project management can be said to begin with control of risk. So, if you are a project manager who is constantly getting burned by out-of-control risks flaming into problems, you may want to consider carefully what Teddy Roosevelt said about risk—you may also want to throw away your book of matches.

Dr. Robert N. Charette is President of the ITABHI Corporation and current chair of the SEI Risk Advisory Board. He can be reached at:
75000.1726@Compuserve.com
or by writing to:
PO Box 1929, Springfield, VA 22151 ■

PLANNING FOR (SOFTWARE) RISK

by Kevin Wall, NYMA, FAA SETA Program

Risk can be defined as, "The probability of an **undesirable** event multiplied by the **significance** (cost) of the consequence should it occur." These are the two elements which are key to risk management. No risk management approach can provide a guarantee against cost and schedule overruns. But proper risk management planning can significantly reduce their probability and/or impact. Programs can prioritize potential risks and plan the implementation of risk mitigation activities. They can initiate activities to reduce the probability of occurrence of and/or implement strategies to reduce the impact of the most significant risks.

Most risks, however categorized, will eventually impact cost and schedule. In today's environment, air traffic systems are primarily dependent on software for successful operation and maintenance. Thus, major problems with software invariably lead to major Program failures—the acquisition and development of software should always be viewed as a high risk activity. Yet software acquisition does not always receive the attention given to hardware acquisition, with predictable results. Recognizing that software use is risky is a first step, but properly identifying, assessing, tracking, and controlling that risk—managing it—are all the other steps. **Please note that using NDI and COTS software is at least as risky as contracting to develop software! The risks are just different.** This is where a well developed Risk Management Plan shows its worth.

The Air Traffic Systems Development Organization (AUA) has published an Acquisition Risk Management Guide and is providing an automated risk management tool, the Risk Management Module (RMM). These can help the Integrated Product Teams in developing a comprehensive and workable Risk Management Plan (RMP), and in managing the identified acquisition risks. The Guide presents a systematic approach to acquisition risk identification and assessment, and describes methodologies to help in implementing a RMP, without which, the Team is truly "flying blind."

The Risk Management Module (RMM) is included in the IPT_Toolset. RMM is used to provide automated tracking of the status of defined risk items. Identified risks and attributes are input to a risk worksheet display which specifies key data elements for status reports. The risk worksheet establishes POCs, detailed risk description, key issues, assessment elements in terms of probability of occurrence and severity of impact, candidate strategies and plans of action, a schedule of risk mitigation actions to be taken, risk parameters and measures to be tracked, and milestones. For example, the tool can establish inchstones and milestones for software inspections as informal (or formal) reviews of individual software activities such as specification, design, code, test case preparation, etc., for critical components down to the unit level.

An automated project scheduling system can track the general aspects of work tasks well, but with a complex system the attributes specifically related to risk tend to be missed. Program risk measures need to be tracked, analyzed, and reported separately from the as-built product delivery, quality, and performance measures. The product measures tell the PM where the product is right now; the risk measures serve as a warning about what can be expected just over the horizon and even further down the road.

A properly implemented risk management tool can be a major step towards keeping Teams informed about potential problems. However, for maximum benefit, the tool output must be used to plan and replan, to alter schedules, to shift resources, and otherwise to manage the acquisition/development process continually to minimize risk of cost and schedule overrun.

Using an Acquisition and Program Risk Management System is an insurance policy against cost and schedule overruns—it is not a guarantee that these will not occur. However, if the system is properly administered, it can promote the timely implementation of mitigation activities to sharply reduce or even altogether avoid impact. Informed oversight by the IPTL is still a major consideration in implementing a viable risk program. As in most management initiatives, if you want to make it happen, total endorsement of the process by all levels of management and the working troops is critical to success.

The Acquisition and Program Risk Management System complements other elements of the IPT acquisition process by providing concepts, methodologies, and tools to assist in developing risk management plans and activities, in maintaining the plans to reflect the most current risk mitigation strategies, in tracking results, and in continually assessing the current and future acquisition Program risk environment. "Risk," as an individual program management element, has to be addressed and reported on as a separate agenda item during program reviews. While acquisition risk management, like any insurance premium, adds to up front cost, in an era in which we cannot afford multibillion dollar gambles, it is mandatory to Program and Agency success. ■

Eye on SPI

by Susan Gardner, AIT-5

This quarter, the SEPG continued work on our Metrics Program, released the ***draft*** COTS/NDI and Open Systems guidelines, planned and held a second ***Partnerships for Improvement with Industry*** meeting, organized the second SEEC meeting, and offered many training courses.

Software Metrics. The metrics working group has recommended doing a pilot using the Practical Software Measures guide already prepared by the Joint Logistics Command. Planning is underway on how the program will be expanded to include multiple projects.

Training. Two classes on use of MIL-STD-498 were offered in June and received rave reviews from attendees. Several additional offerings will be held in the future. Classes in CMM, Cost Estimation, SCEs and Open Systems were also held this quarter.

COTS/NDI Guidelines and Open Systems Guidelines. The ***draft*** COTS/NDI Guidelines were released to the SEPG for comment in June, and the ***draft*** Open Systems Guidelines in July. The final versions are planned for the end of September.

SEEC Meeting. In June, the SEEC met for the second time. Three presentations were given. Cindy King briefed on the local AUA SEPG and its activities. Susan Gardner briefed on the six FAA SEPG strategies. Lloyd Mosemann, former Deputy Assistant Secretary of Defense, briefed on his experiences in DOD when he instituted software quality improvement initiatives. The SEEC members were pleased with the progress of the SEPGs, and agreed with the proposed strategies. Task plans are being developed to implement these six strategies.

Partnerships for Improvement with Industry Meeting. On July 22nd, the second meeting with the Industry Partners was held at FAA headquarters. The subject for the day was SCEs. FAA is specifically seeking input from industry as we prepare FAA Guidelines for use of SCEs. After a presentation by Peter Challan, AUA-2, three breakout sessions were held: ***Using SCEs for Supplier Selection, Contract Monitoring, and Baseline Performance***; ***Benefits & Pitfalls of CMM-Based Appraisals—Industry Experiences***; and ***Multiple Use of CMM-Based Appraisals***. Detailed proceedings of the discussions will be published separately.

Please continue to keep up with and support the activities of **your** FAA SEPG. Contact your FAA SEPG representative if you are interested in participating in any way. ■

"Engineering Disasters..." continued from page 1

but one has failed at it, one has performed acceptably, and one has performed outstandingly. Clearly, depending on the skill pool and if other priorities permit, the choice is to one of the latter two types. If the task is sufficiently critical, the choice may be only to someone of the last type. The PM must give extra special care to the selection of his chief engineer, and plan to accommodate her strengths and weaknesses. (A good chief engineer will devise a similar plan to accommodate her PM.)

If only people of the first two types are available, the manager must spend considerably more effort in deciding among the candidates. But if a manager lacks key people for a Program, she must have the courage to tell her management and insist on a workable solution. Agreeing to "take on" an impossible Program is managerial malpractice. Needless to say, it is impossible to categorize people with respect to a job if the manager has no notion of what skills the proposed job requires, or what skills jobs previously performed by the candidates required. Both are easily determined, as well as the performance of engineers on past jobs, but not without effort on the manager's part.

In deciding to give someone "another chance," the manager must assess whether that someone has demonstrated an unremediated lack of some necessary technical or personality skill. In determining if someone deserves a crack at something that will probably stretch his abilities, the manager must decide whether that person has the basic technical and personality skills needed, and is likely to rise to the occasion. In either case, it must be assumed that an individual that has failed at or never performed a task before will need extra support in the way of training, mentoring, and supervision. That person is also likely to take longer to perform the task than one of the successful people. Due allowance must be made.

The second line of defense is to ensure that the engineer assigned to the task has the necessary support structure and resources. If these are not supplied in sufficient quality and quantity, the correlation between past and future success is bound to be poor. No matter how skillful the carpenter, she is unlikely to do much of a job of nailing one board to another without a hammer or nails, or if one board is at another site two miles away. Unfortunately, it is common to overload an outstanding performer. It is easy to assume that even a fraction of one is better than all of a weak performer. Or, that a strong performer can "make do" with significantly less (i.e., inadequate) support and/or significantly fewer resources. Amost equally disastrous is the practice of rationing resources "impartially" with little or no regard to the difficulty of the job or the ability of the performer. It doesn't work for parents and it won't work for managers.

A third line of defense is planned redundancy. It always amazes me how top managers will skimp on a Program when sufficient planning and use of resources will ensure delivery, then spend, seemingly without limit, once failure is imminent and generally unavoidable. If you plan for failure up front, it is avoidable. Suppose the only candidates for a job are of the first two types. Then start two, or even three, on different aspects of the same job, in parallel. When one or two of your candidates fail, go with the successful candidate. Never, ever, make the mistake of shifting resources from a succeeding candidate to a failing candidate. This is a formula for ensuring that everyone will fail which has been proved over and over in military combat. Cut your losses quickly—and this includes getting rid of bad engineers quickly. (A bad engineer is one that is not only incompetent, but takes no responsibility for, and therefore does not learn from, the negative consequences of his actions.)

If you wind up with all failing candidates, make sure you have the machinery in place to detect this early and, as each candidate fails, switch to an alternative strategy. This may involve supporting or replacing them with a more successful performer. It is one reason why you want to plan to support the weak candidate with more time and resources and with more supervision at the outset. It is also a reason never to start a Program without some engineering reserve (which may be no more than potential overtime). Programs which start with everyone already overcommitted always fail. For critical tasks, it is never a bad idea to back up even the best engineers. Consider this "bus" insurance. (You must always worry about and plan for your key engineers getting hit by a bus or other catastrophe.)

Using a low risk, redundant approach is also a way to insure against the failure of any high risk approach. The resources expended on the redundant alternative approach(es) should be considered as an insurance premium, even if unnecessary. Where a parallel approach strategy is used, multiple successes will speed the result, so not much is lost if properly planned for.

There are dozens of other strategies a competent manager can use to insure against bad engineering or other disasters. Enough should be used to reduce the risk of failure to a tolerable level at an acceptable cost. ■

Training Opportunities...

The FAA SEPG has developed a training program consisting of the following topics. Classes are to be offered periodically throughout the year. Please contact your organization's SEPG member for schedule and enrollment information or discussion of your software training needs.

- **Capability Maturity Model (CMM) for Software and Associated Key Process Areas for Level 2**
- **CMM for Software Acquisition and Associated Key Process Areas for Level 2**
- **People CMM**
- **Defining Software Processes**
- **Consulting Skills Workshop**
- **Open Systems, the Promises and the Pitfalls**
- **Software Capability Evaluation Training**
- **Software Risk Management**
- **Requirements Management**
- **Metrics**
- **Clean Room**
- **Cost Estimation and Economic Evaluation of Projects**
- **MIL-STD-498, Use and Tailoring Opportunities**

In This Issue

interFACE

Newsletter of the Software Engineering Process Group

Volume 5, Number 3
August 1996

SEPG
FEDERAL AVIATION ADMINISTRATION

Conference Calendar

Society for Software Quality (SSQ) Meeting
Held Monthly
Future Monthly Topics:
September 9 Why Not Have a Software Process Assessment?
October 8 Software Enbgineering Technology: Year in Review
Contact: Chris Dryer (202) 767-2894

SEI Software Engineering Symposium
September 9-12
Pittsburgh, PA
Contact: SEI Customer Relations (412) 268-5800

30th Annual Engineering & Technical Management Workshop
September 30-October 4
Baltimore, MD
Contact: Adrienne Scott (202) 651-2243

Software Process Improvement Network (SPIN) Meeting
Held Monthly
Future Monthly Topics:
September 4 Acquisition Risk Management
October 2 Inside an Effective Analysis Process
Contact: Jonathan Addelston (703) 848-6530

Society for Software Quality (SSQ) Meeting
Held Monthly
Contact: Chris Dryer (202) 767-2894

Federal Software Process Improvement Working Group (FEDSPIWG)
Held Monthly at NOAA
Contact: Martha Morphy at NOAA (301) 713-3345

DOT/FAA/AIT-5
800 Independence Avenue, SW
Washington, DC 20591